

**REMARKS**

The Office Action of December 4, 2001 made a statutory double patenting rejection of claim 1 under 35 U.S.C. § 101, stating that claim 1 recites the same invention as claim 1 of U.S. Patent No. 6,259,442. By this paper, claim 1 has been amended and new claims 2-29 have been added. Accordingly, claims 1-29 are pending. Applicants respectfully request favorable reconsideration of the pending claims in view of the amendments made herein.

By this paper, claim 1 has been amended so that it is no longer coextensive in scope with claim 1 of U.S. Patent No. 6,259,442. Applicants further submit that new independent claims 12, 16 and 25 are also not coextensive in scope with any of the claims of the '442 patent. Thus, the statutory double patenting rejection of claim 1 has been overcome and Applicants respectfully request substantive examination of the pending claims.

Applicants also note that an information disclosure statement was filed on July 31, 2001. Applicants request that the Examiner initial the a copy of the form PTO-1449 filed with the information disclosure statement and return the same to the Applicants with the next action.

Attached hereto is a marked-up version of the changes made to the previous version of the specification and claims by this amendment. The attached pages are captioned "VERSION WITH MARKINGS TO SHOW CHANGES MADE."

For the foregoing reasons, Applicants submit that the pending claims are in condition for allowance and courteously request favorable action. If there are any outstanding issues that could be resolved by telephone, the Examiner is invited to contact the undersigned attorney.

Dated this 4<sup>th</sup> day of March, 2002.

Respectfully submitted,



R. BURNS ISRAELSEN  
Attorney for Applicant  
Registration No. 42,685

WORKMAN, NYDEGGER & SEELEY  
1000 Eagle Gate Tower  
60 East South Temple  
Salt Lake City, Utah 84111  
Telephone: (801) 533-9800  
Facsimile: (801) 328-1707



**022913**

PATENT TRADEMARK OFFICE

RBI:llr

G:\DATA\WPDOCSRN\WEBTV\OTHERDOC\5.1.5 amendment a.DOC

**VERSION WITH MARKINGS TO SHOW CHANGES MADE**

**IN THE CLAIMS**

Claim 1 has been amended as follows:

1. (Amended) In a client system that communicates with [computer system that includes a plurality of client systems and] a [plurality of] server system [systems, all interconnected by a network infrastructure, wherein the plurality of server systems provide access to sites storing Web pages or other data], [and] wherein [one or more of] the client system includes [systems comprises] a computing device that includes program instructions used in the operation of the [one or more] client system [systems], a method of [automatically and without user intervention] restoring a corrupted portion of the program instructions at the client system, [the method] comprising [the client system performing] the acts of:

[during initialization,] checking the validity of system program instructions at the client system and [thereafter] checking the validity of application program instructions at the client system to determine whether the system program instructions or the application program instructions have a corrupted portion [; determining that a corrupted state exists in at least a portion of any of the checked program instructions];

if it is determined [in response to determining] that either the system instructions or the application program instructions have a corrupted portion [state exists], connecting the client system to [one of] the server system [systems or sites that contains a replacement for the corrupted portion of the program instructions];

receiving [the] replacement instructions from [said one of] the server system [systems or sites]; and

replacing the corrupted portion of the program instructions with the replacement instructions.

New claims 2-29 have been added as follows:

2. (New) A method as recited in claim 1, wherein the act of checking the validity comprises the act of using a checksum technique to determine whether the corrupted portion exists.

3. (New) A method as recited in claim 1, wherein the act of connecting the client system to the server system comprises the acts of:

selecting a local connection script associated with the server system; and  
connecting to the server system using the local connection script.

4. (New) A method as recited in claim 3, wherein the act of selecting a local connection script associated with the server system comprises the act of:

reading a default connection script from a memory of the client system;  
using the default connection script to connect to remote computer that contains the selected local connection script; and  
downloading the selected local connection script from the remote computer.

5. (New) A method as recited in claim 1, wherein the act of receiving replacement instructions from the server system comprises the act of automatically, and without user intervention, requesting the replacement instructions from the server system after connecting to the server system.

6. (New) A method as recited in claim 5, wherein the act of receiving replacement instructions from the server system comprises receiving replacement instructions that have been downloaded from the server system over a satellite link that connects the server system with the client system.

7. (New) A method as recited in claim 5, further comprising the acts of:  
writing the replacement instructions to a random access memory of the client system;  
decompressing the replacement instructions; and

writing the decompressed replacement to a flash memory of the client system.

8. (New) A method as recited in claim 1, wherein the replacement instructions are received from the server system over the Internet.

9. (New) A method as recited in claim 8, wherein the application program instructions comprise an Internet browser, and wherein the corrupted portion is included in Internet browser.

10. (New) A method as recited in claim 1, wherein the replacement instructions for the corrupted portion of the program instructions comprise a Java applet.

11. (New) A method as recited in claim 1, wherein the act of checking the validity of the system program instructions is performed during initialization.

12. (New) A computer program product for implementing, in a client system that communicates with a server system, wherein the client system includes a computing device that includes program instructions used in the operation of the client system, a method of restoring a corrupted portion of the program instructions at the client system, the computer program product comprising:

a computer-readable medium carrying computer-executable instructions, that when executed at the client system, cause the client system to perform the method, including the acts of :

    checking the validity of system program instructions at the client system and checking the validity of application program instructions at the client system to determine whether the system program instructions or the application program instructions have a corrupted portion;

    if it is determined that either the system instructions or the application program instructions have a corrupted portion, connecting the client system to the server system;

    receiving replacement instructions from the server system; and

    replacing the corrupted portion of the program instructions with the replacement instructions.

13. (New) A computer program product as recited in claim 12, wherein act of checking the validity comprises the act of using a checksum technique to determine whether the corrupted portion exists.

14. (New) A computer program product as recited in claim 12, wherein the method performed by the client system further comprises the acts of:

    writing the replacement instructions to a random access memory of the client system;

    decompressing the replacement instructions; and

    writing the decompressed replacement instructions to a flash memory of the client system.

15. (New) A computer program product as recited in claim 12, wherein the act of checking the validity of application program instructions is performed upon the client system establishing communication with the server system in preparation for requesting from the server system information other than the replacement instructions.

16. (New) In a client system that communicates with a server system, wherein the client system includes a computing device that includes program instructions used in the operation of the client system, a method of restoring a corrupted block of the program instructions at the client system, comprising the acts of:

checking the validity of blocks of the program instructions at the client system to determine whether the blocks of the program instructions include a corrupted block;

if it is determined that the blocks of the program instructions include a corrupted block, requesting a replacement block of program instructions from the server system;

requesting the replacement block of program instructions from the server system;

receiving the replacement block of program instructions from the server system;  
and

replacing the corrupted block with the replacement block of program instructions.

17. (New) A method as recited in claim 16, wherein the act of checking the validity comprises the act of using a checksum technique to determine whether the corrupted block of the program instructions exists.

18. (New) A method as recited in claim 16, wherein the act of requesting comprises the act of connecting the client system to the server system over the Internet.

19. (New) A method as recited in claim 16, wherein the act of requesting comprises the acts of:

selecting a local connection script associated with the server system; and  
connecting to the server system or sites using the local connection script.

20. (New) A method as recited in claim 19, wherein the act of selecting a local connection script comprises the acts of:

reading a default connection script from a memory;

using the default connection script to connect to a remote computer that contains the selected local connection script; and

downloading the selected local connection script from the remote computer.

21. (New) A method as recited in claim 16, wherein the act of receiving the replacement block from the server system comprises the act of receiving the replacement block that has been downloaded from the server system over a satellite link that connects the server system with the client system.

22. (New) A method as recited in claim 16, each of the blocks of the program instructions being stored at a specific storage address at the client system, wherein the act of checking the validity of blocks of the program instructions at the client system comprises the act of identifying the specific storage address associated with any corrupted block that is identified.

23. (New) A method as recited in claim 16, further comprising the acts of:  
writing the replacement block to a random access memory of the client system;  
decompressing the replacement block; and  
writing the decompressed replacement block to a flash memory of the client system.

24. (New) A method as recited in claim 17, wherein the act of requesting the replacement block of program instructions from the server system is performed automatically and without user intervention.

25. (New) A computer program product for implementing, in a client system that communicates with a server system, wherein the client system includes a computing device that includes program instructions used in the operation of the client system, a method of restoring a corrupted block of the program instructions at the client system, the computer program product comprising:

a computer-readable medium carrying computer-executable instructions that when executed at the client system, cause the client system to perform the method, including the acts of:

    checking the validity of blocks of the program instructions at the client system to determine whether the blocks of the program instructions include a corrupted block;

    if it is determined that the blocks of the program instructions include a corrupted block, requesting a replacement block of program instructions from the server system;

    requesting the replacement block of program instructions from the server system;

    receiving the replacement block of program instructions from the server system; and

    replacing the corrupted block with the replacement block of program instructions.

26. (New) A computer program as recited in claim 25, wherein the act of checking the validity comprises the act of using a checksum technique to determine whether the corrupted block of the program instructions exists.

27. (New) A computer program product as recited in claim 25, wherein the method performed by the client system further comprises the acts of:

    writing the replacement block to a random access memory of the client system;

    decompressing the replacement block; and

    writing the decompressed replacement block to a flash memory of the client system.

28. (New) A computer program product as recited in claim 25, wherein the act of checking the validity of blocks of the program instructions is performed upon the client system establishing communication with the server system in preparation for requesting from the server system information other than the replacement block.

29. (New) A computer program product as recited in claim 25, each of the blocks of the program instructions being stored at a specific storage address at the client system, wherein the act of checking the validity of blocks of the program instructions at the client system comprises the act of identifying the specific storage address associated with any corrupted block that is identified.